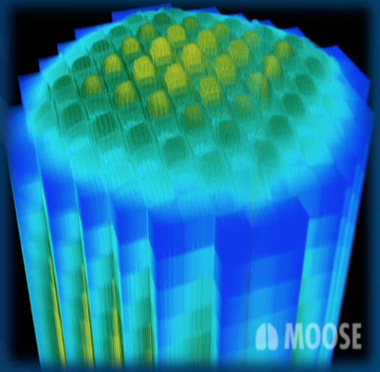
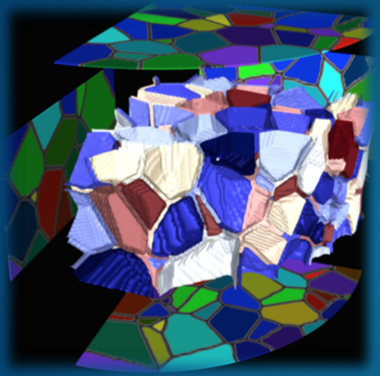
MOOSE

An open-source framework to enable rapid development of collaborative, multi-scale, multi-physics simulation tools





Andrew E. Slaughter

Team Members: Derek Gaston, Cody Permann, David Andrš, John Peterson, Jason Miller

www.inl.gov

Idaho National Laboratory

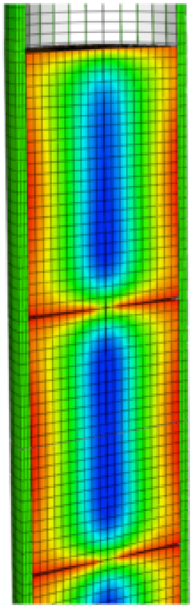
MOOSE Overview

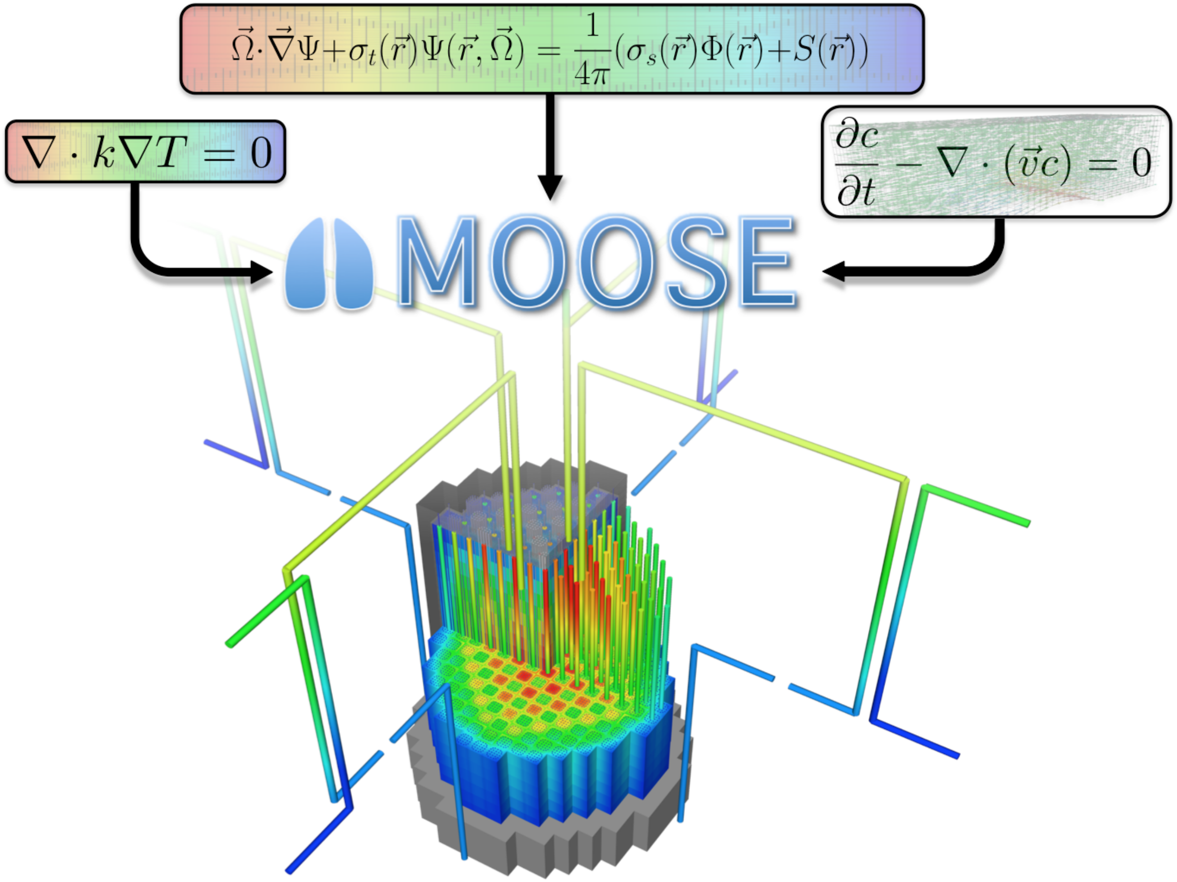
MOOSE Team

- Derek Gaston
  - derek.gaston@inl.gov
  - @friedmud
- Cody Permann
  - cody.permann@inl.gov
  - @permcody
- David Andrš
  - david.andrs@inl.gov
  - @andrsdave
- John Peterson
  - jw.peterson@inl.gov
  - @peterson512
- Jason Miller
  - jason.miller@inl.gov
  - @mjmiller96
- Andrew Slaughter
  - andrew.slaughter@inl.gov
  - @aeslaughter98

MOOSE: Multiphysics Object Oriented Simulation Environment

- A framework for solving computational engineering problems in a well-planned, managed, and coordinated way
  - *Leveraged across multiple programs*
- Designed to significantly reduce the expense and time required to develop new applications
- Designed to develop analysis tools
  - *Uses robust solution methods*
  - *Designed to be easily extended and maintained*
  - *Efficient on both a few and many processors*





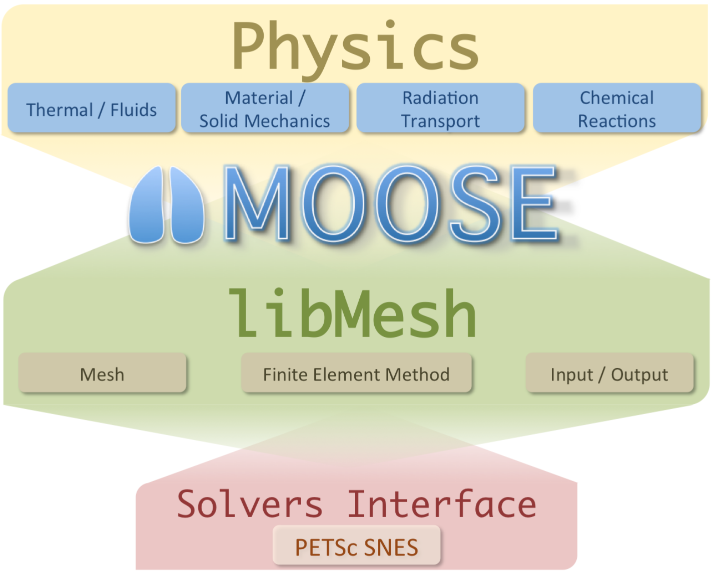
$\vec{\Omega}\cdot\vec{\nabla}\Psi+\sigma_t(\vec{r})\Psi(\vec{r},\vec{\Omega}) = \frac{1}{4\pi}(\sigma_s(\vec{r})\Phi(\vec{r})+S(\vec{r}))$
$\nabla \cdot k\nabla T = 0$
$\frac{\partial c}{\partial t} - \nabla \cdot (\vec{v}c) = 0$
MOOSE

Capabilities

- 1D, 2D and 3D
  - User code agnostic of dimension
- Finite Element Based
  - Continuous and Discontinuous Galerkin (and Petrov Galerkin)
- Fully Coupled, Fully Implicit
- Unstructured Mesh
  - All shapes (Quads, Tris, Hexes, Tets, Pyramids, Wedges, ...)
  - Higher order geometry (curvilinear, etc.)
  - Reads and writes multiple formats
- Mesh Adaptivity
- Parallel
  - User code agnostic of parallelism
- High Order
  - User code agnostic of shape functions
  - p-Adaptivity
- Built-in Postprocessing
- And much more ...

Code Platform

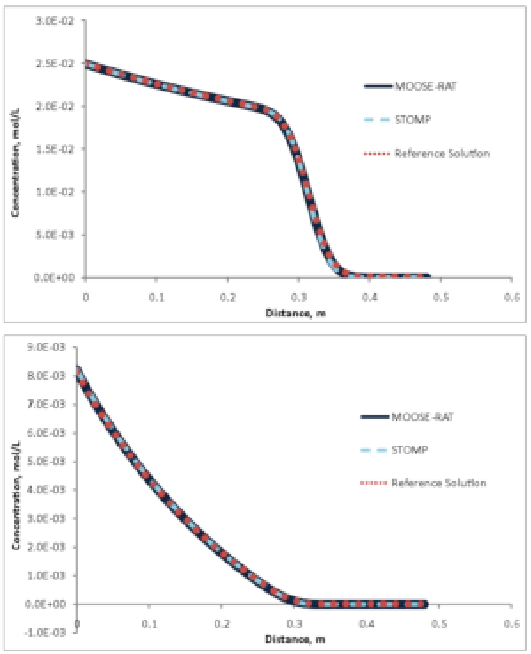
- MOOSE is a simulation framework allowing rapid development of new simulations tools.
- Specifically designed to simplify development of multiphysics applications.
- Provides an object-oriented, pluggable system for defining all aspects of a simulation tool.
- Leverages multiple DOE developed scientific computational tools
- Allows scientists and engineers to develop state of the art simulation capabilities.
- **Maximized Science/$**
- Currently ~64,000 lines of code.



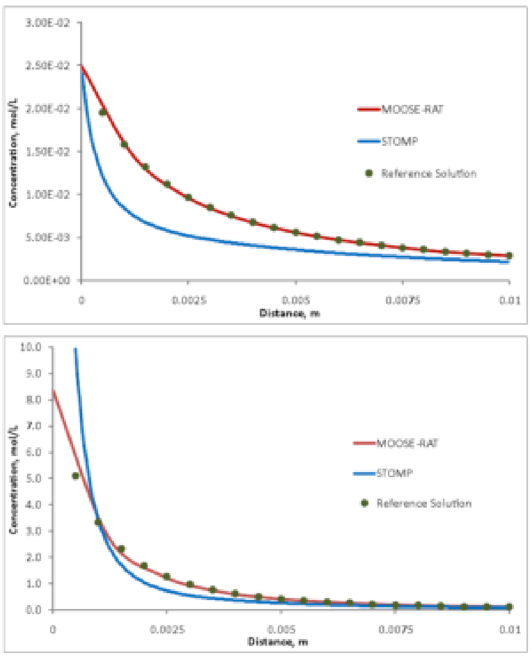
Solves Problems that Challenges Others

Strong Coupling: better agreement between MOOSE and reference **Weak Coupling:** excellent agreement between MOOSE and operator-split

Profile of A concentration at 4480 sec.



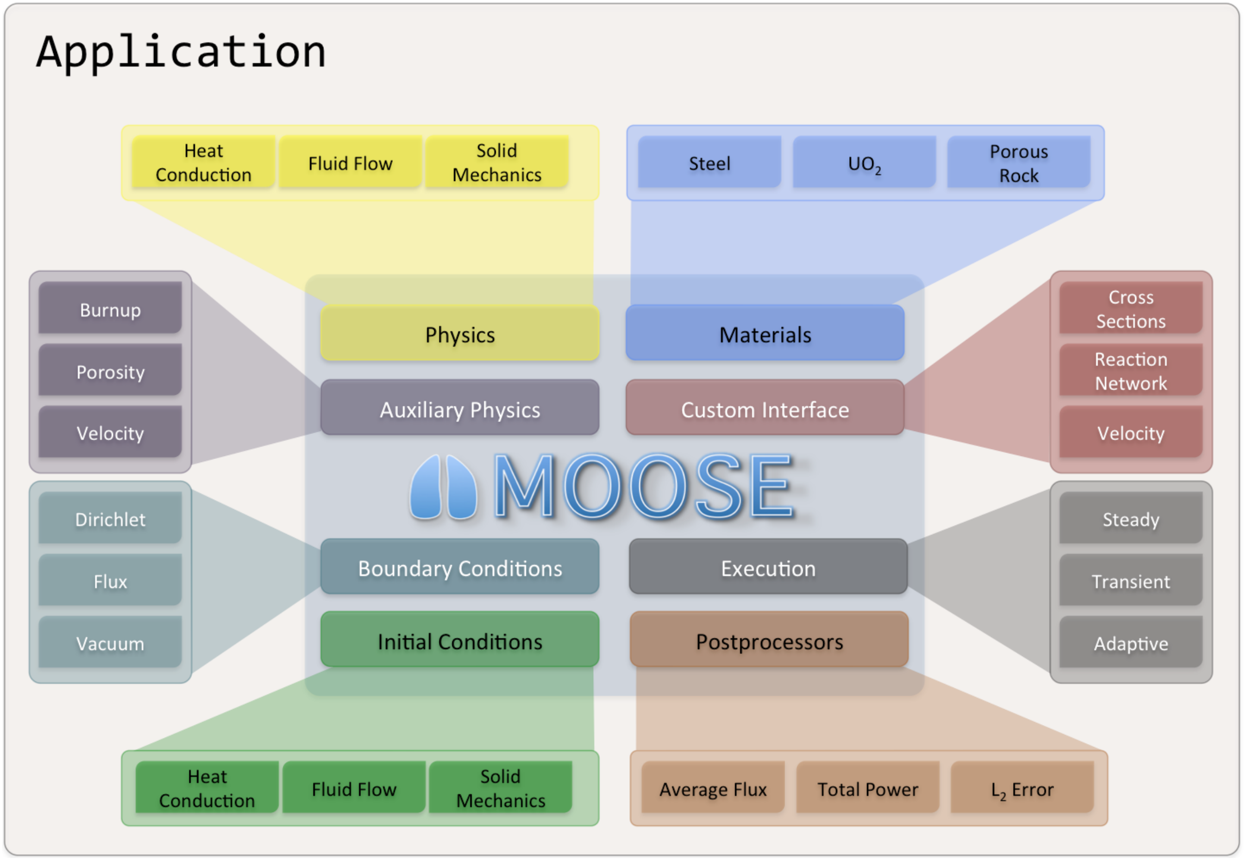
Profile of C concentration at 4480 sec.



Rapid Development

Application	Physics	Results	Lines
BISON	Thermo-mechanics, Chemical, diffusion, coupled mesoscale	4 mo.	3,000
PRONGHORN	Neutronics, Porous flow, eigenvalue MARMOT 4th order phasefield mesoscale	3 mo.	7,000
MARMOT	4-th order phase-field meso-scale	1 mo.	6,000
RAT	Porous ReActive Transport	1 mo.	1,500
FALCON	Geo-mechanics, coupled mesoscale	3 mo.	3,000
MAMBA	Chemical reactions, prescipitation, and porous flow	5 wks.	2,500
HYRAX	Phase-field, ZrH precipitation	3 mo.	1,000
PIKA	Multi-scale heat and mass transfer with phase-change	3 mo.	2,900

MOOSE Application Architecture



MOOSE Code Example

Strong Form

$$\rho Cp\frac{\partial T}{\partial t} - \nabla \cdot k(T,B)\nabla T = f$$

Weak Form

$$\int_\Omega \rho Cp\frac{\partial T}{\partial t}\psi_i + \int_\Omega k\nabla T\cdot\nabla\psi_i - \int_{\partial\Omega} k\nabla T\cdot\mathbf{n}\psi_i - \int_\Omega f\psi_i = 0$$

Kernel Kernel BoundaryCondition Kernel

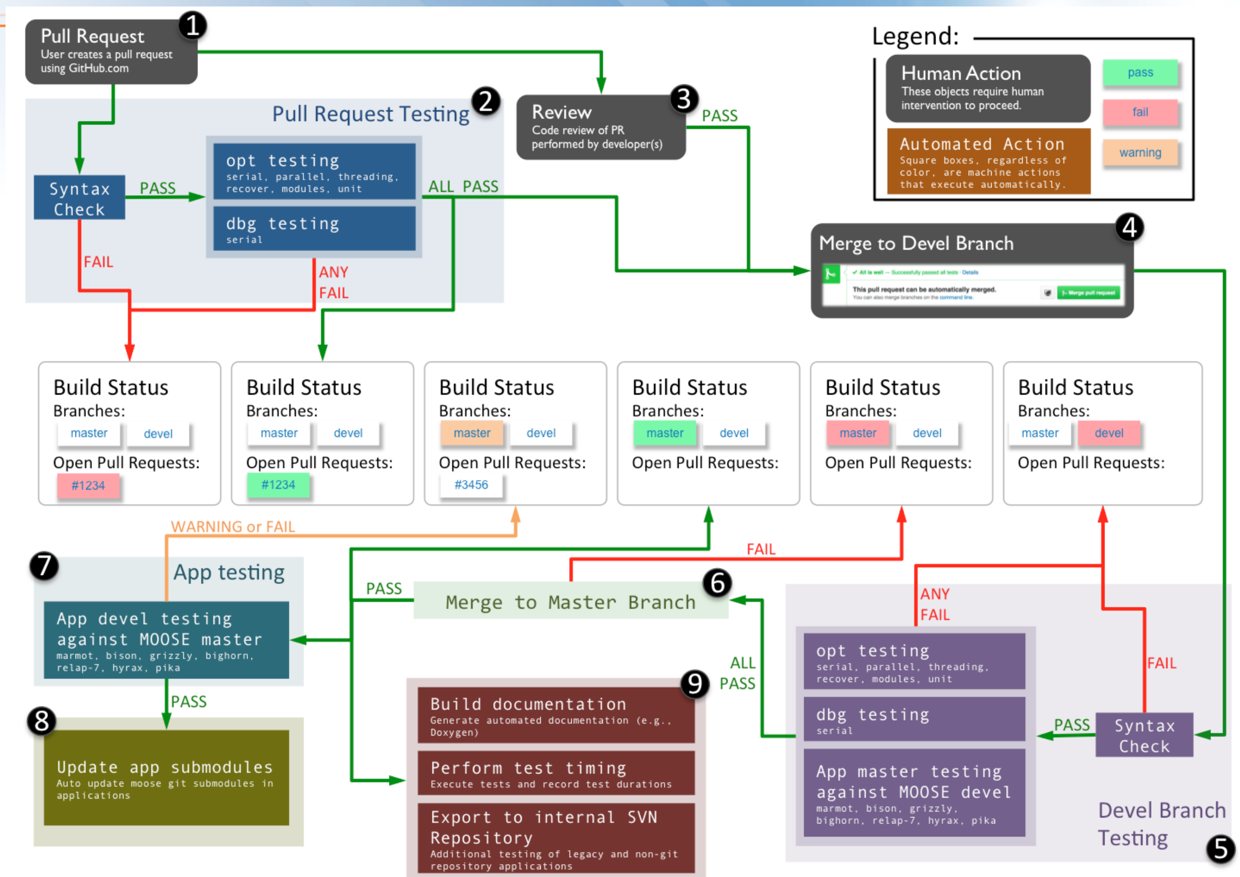
Actual Code

```
return _k[_qp]*_grad_u[_qp]*_grad_test[_i][_qp];
```

MOOSE Software Quality Practices

- MOOSE currently meets all NQA-1 (Nuclear Quality Assurance Level 1) requirements
- All commits to MOOSE undergo review using GitHub Pull Requests and must pass a set of application regression tests before they are available to our users.
- All changes must be accompanied by issue numbers and assessed an appropriate risk level.
- We maintain a regression test code coverage level of 80% or better at all times.
- We follow strict code style and formatting guidelines (wiki/CodeStandards).
- We monitor code comments with third party tools for better readability.

tional Laboratory



Modules

Modules Overview

MOOSE includes a set of community developed physics modules that you can build on to create your own application.

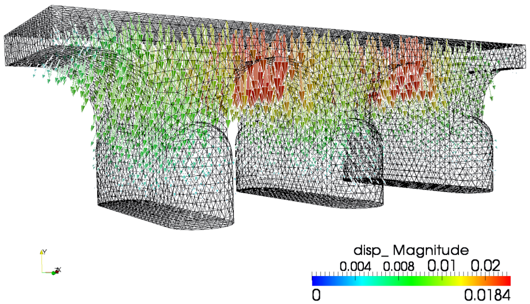
- Phase Field
- Tensor Mechanics
- Heat Conduction
- Multiphase flow through porous media (Richards Eq.)
- Chemical Reactions

Modules Overview (cont.)

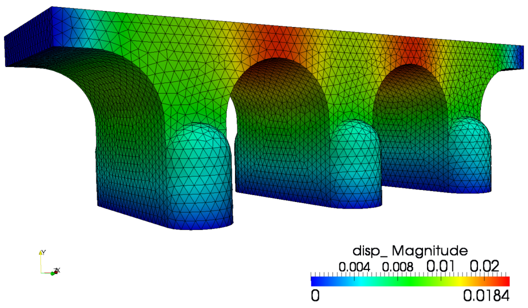
- The purpose of the modules is to encapsulate common kernels, boundary conditions, etc. to prevent code duplication.
- Examples include: heat conduction, solid mechanics, Navier-Stokes, and others.
- *No* export controlled physics (e.g., neutronics) should be put into the modules.
- The modules are organized so that your application can link against only those which it requires.

Modules: Solid Mechanics Example

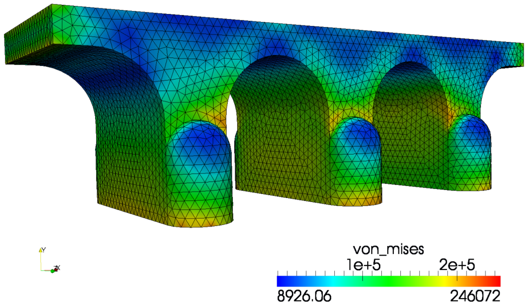
- Available in: [moose/modules/solid_mechanics]()
- Stats:
  - 127,650 elements, 25,227 nodes
- Features:
  - Large displacements
  - Plasticity and Creep



Displacement



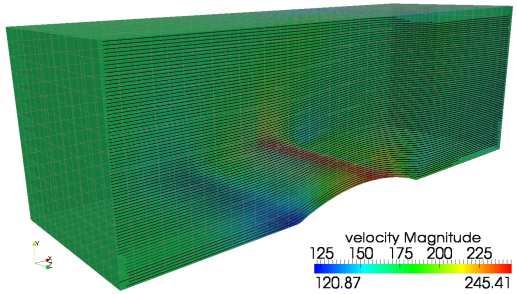
Displacement Magnitude



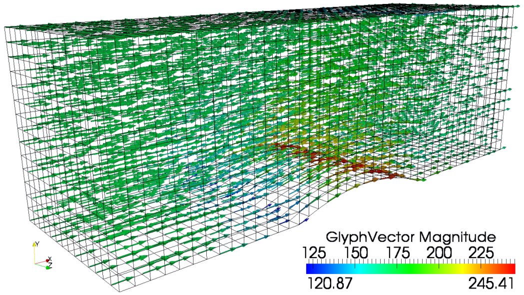
Von Misses Stress

Modules: Flow Example

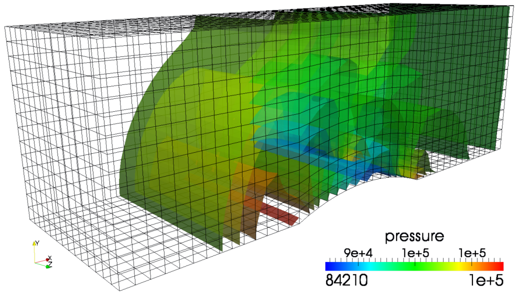
- Available in: moose/modules/navier_stokes
- Subsonic Test Case:
    - Mach 0.5 over a circular arc
    - Euler equations
    - 8,232 elements, 9,675 nodes



Velocity Streamlines



Velocity



Pressure

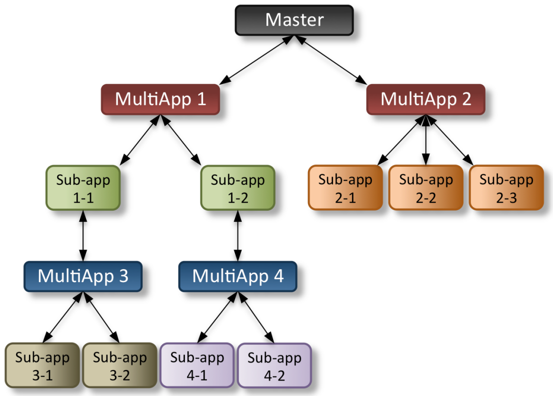
MultiApps

MultiApps Overview

- MOOSE was originally created to solve fully-coupled systems of PDEs.
- Not all systems need to be / are fully coupled:
    - Multiscale systems are generally loosely coupled between scales.
    - Systems with both fast and slow physics can be decoupled in time.
    - Simulations involving input from external codes might be solved somewhat decoupled.
- To MOOSE these situations look like loosely-coupled systems of fully-coupled equations.
- A `MultiApp` allows you to simultaneously solve for individual physics systems.

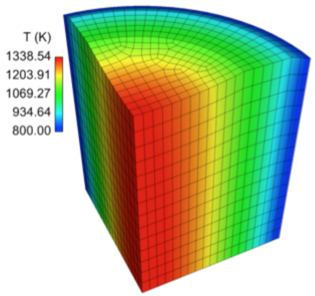
MultiApps Overview (cont.)

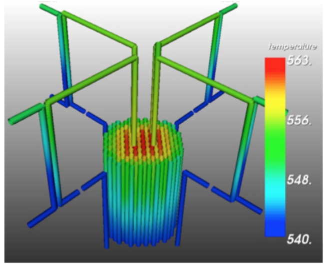
- Each "App" is considered to be a solve that is independent.
- There is always a "master" App that is doing the "main" solve.
- A "master" App can then have any number of `MultiApps`.
- Each `MultiApp` can represent many (hence Multi!) "sub-apps".
- The sub-apps can be solving for completely different physics from the main application.
- They can be other MOOSE applications, or might represent external applications.
- A sub-app can, itself, have `MultiApps`... leading to multi-level solves.

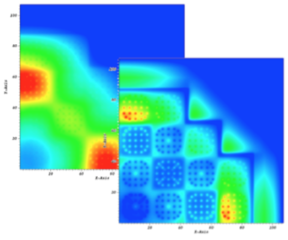


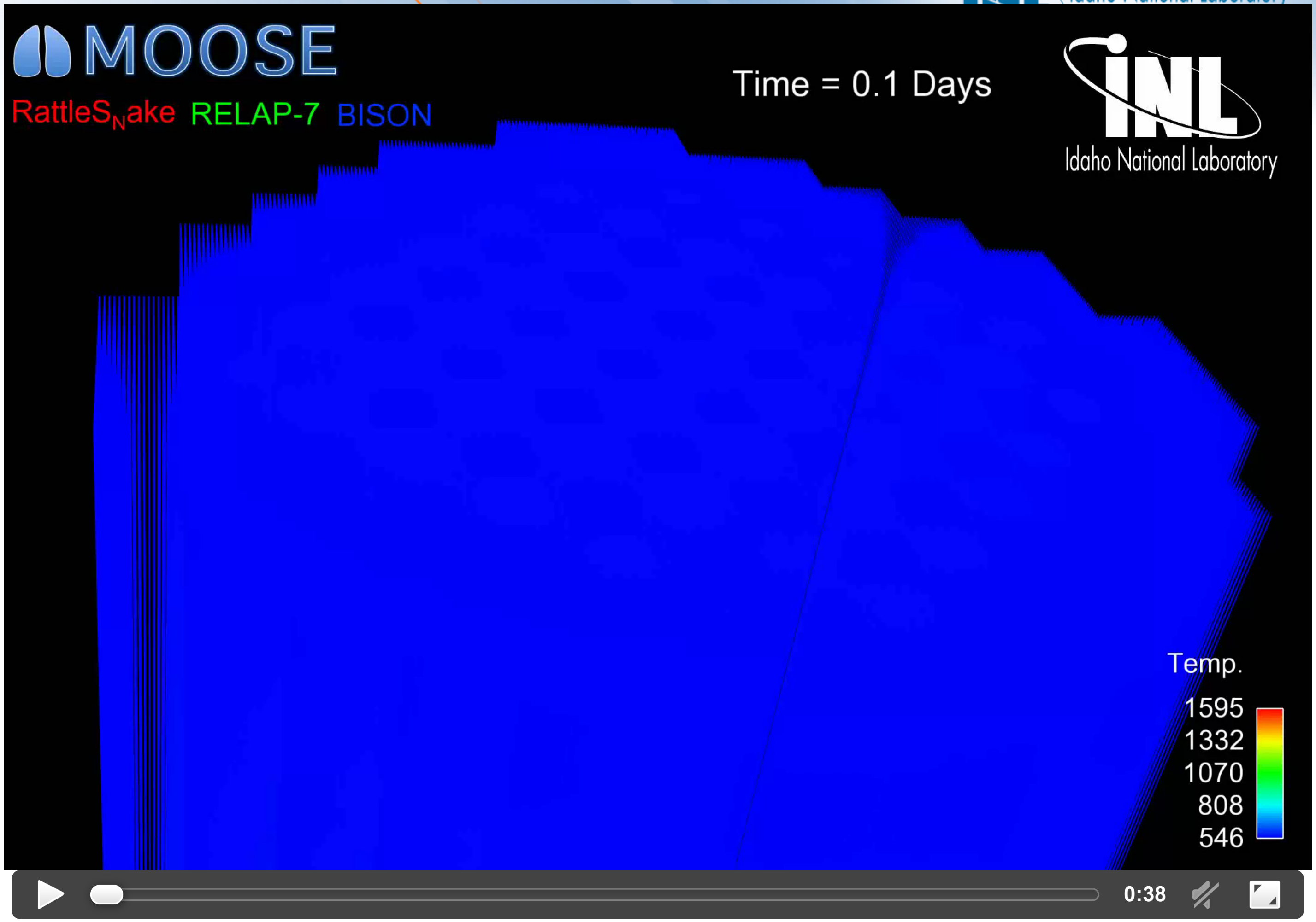
MultiApp Demonstration

- BISON Fuels Performance Application
  - Solves the fully-coupled thermomechanics and species diffusion equations in 2D or 3D
  - Couples to lower length scale material models
  - All fuels capability, LWR, MOX, metal, TRISO, etc.
- RELAP-7 Reactor Systems Analysis
  - Next generation reactor systems analysis application
  - Advanced numerical integration schemes
  - Advanced flow and core physics models
- RattleSNake Multiscale Neutronics Application
  - SAAF: Self-Adjoint Angular Flux formulation
  - SN and PN in flying direction; multi-group in energy; method of lines in time
  - Includes Low-order multi-group diffusion





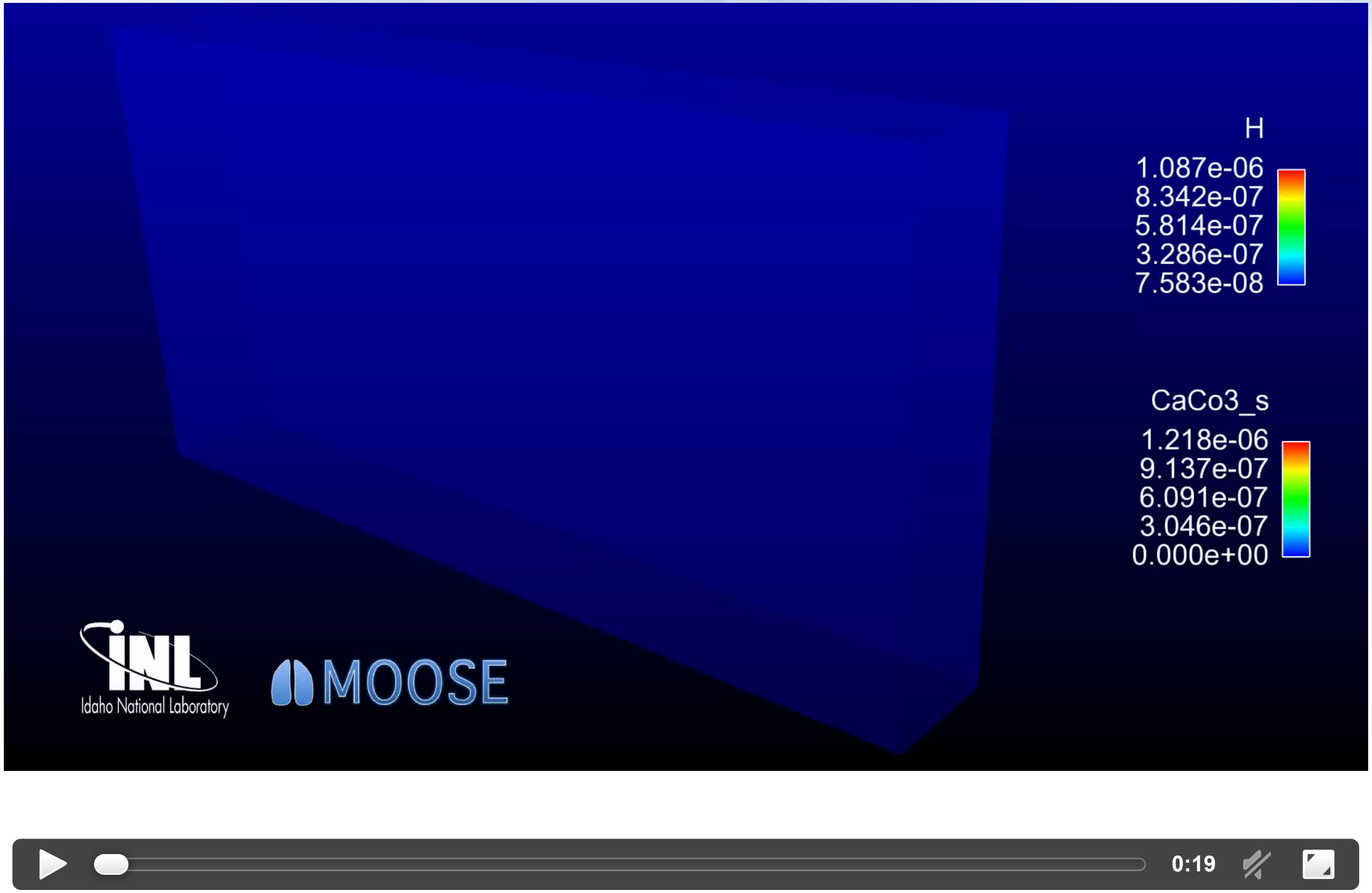



MOOSE
RattleSnake RELAP-7 BISON
Time = 0.1 Days
Idaho National Laboratory
Temp.
1595
1332
1070
808
546
0:38

MOOSE Examples

MOOSE Examples

- Currently, 92 forks of Stork
- Over a dozen applications under heavy development at INL
- Geomechanics projects include:
  - FALCON
  - BADGER
  - RAT
  - RedBack
  - Richards MOOSE module


H
1.087e-06
8.342e-07
5.814e-07
3.286e-07
7.583e-08
CaCo3_s
1.218e-06
9.137e-07
6.091e-07
3.046e-07
0.000e+00
INL
Idaho National Laboratory
MOOSE
0:19

www.mooseframework.org